CONVOLUTIONAL NEURAL NETWORK LAYERS AND ARCHITECTURES

Timea Bezdan*,
Nebojša Bačanin Džakula

Singidunum University,
Belgrade, Serbia

Abstract:
In recent years, computer vision which is one of the fastest growing artificial intelligence disciplines, has become increasingly important in our society due to its wide range applications in different areas such as health care and medicine (algorithms that can diagnose medical images for diseases), vision-based robotics, self-driving cars (that can see and drive safely). Convolutional neural networks are biologically inspired architectures and represent the core of deep learning algorithms in computer vision. In this paper, we represent the fundamental building blocks of convolutional neural networks and the most popular convolutional neural network architectures in the history, including those that have achieved the state-of-the-art performance on standard recognition datasets and tasks such as ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). ILSVRC is one of the largest challenges in computer vision organized by Stanford Vision Lab since 2010 and every year teams compete to claim the state-of-the-art performance on the dataset.

Keywords:
computer vision, deep learning, image recognition, image classification, object recognition.

1. INTRODUCTION

Convolutional neural networks [1] (CNNs) are a specific type of artificial neural networks (ANNs), that has been demonstrated high performance on various visual tasks, including image classification, image segmentation [2], image retrieval [3], object detection [4], image captioning [5], face recognition [6], pose estimation [7], traffic sign recognition [8], speech processing [9], neural style transfer [10] etc.

Convolutional neural networks have become a rapidly growing area of interest in recent years, however, its development started much earlier. One of the most influential papers in this area was published by Hubel and Wiesel in 1959 [11]. They did a series of experiments, trying to understand how neurons work in the visual cortex. The researchers discovered that the visual cortex has a hierarchical organization, that there are simple and complex neurons in the primary visual cortex and that visual processing always starts with simple structures such as oriented edges, complex cells received input from lower level simple cells by

Correspondence:
Timea Bezdan

e-mail:
timea.bezdan.17@singimail.rs

DOI: 10.15308/Sinteza-2019-445-451

way of a receptive field. In 1980, Fukushima introduced Neocognitron [12], which was the first example of an artificial neural network model, that had an idea of simple and complex cells, discovered by Hubel and Wiesel. Fukushima put "S-cells" and "C-cells" into alternative layers, building up into a hierarchy, so-called "sandwich layers" (SCSCS…). S and C cells show similar characteristics to simple and complex cells in the visual cortex. "S-cells" has modifiable parameters and on the top of them, "C-cells" perform pooling. The network was not back-propagated at that time. LeCun in 1989 applied back-propagation [13] to train Fukushima's artificial neural network, the method has a 1% error rate and about 9% reject rate on zip code digits. In 1998, LuCun further optimized CNN using an error gradient-based learning algorithm [1]. In 2012 is proposed AlexNet [14], which has a more complex architecture, it was the first deep convolutional neural network (DCNN). AlexNet [14] achieved significant results and this success has brought about a revolution in computer vision. The significant results came from the efficient use of GPUs, ReLU [15] activation function, regularization technique called dropout [16] and data augmentation.

CNNs are designed to process data that come in the form of multiple arrays, for example, a color image composed of three 2D arrays containing pixel intensities in the three-color channels. They use their convolutional filters to extract information from images, earlier layers detect edges, later layers can detect part of objects, then even later layers may detect complete objects, such as faces, or other complex geometrical shapes [17]. The CNN composed by a set of layers that can be grouped by their functionalities, three main types of layers are: convolutional layer, pooling layer, and fully-connected layer.

2. CONVOLUTIONAL NEURAL NETWORK LAYERS

Convolutional layer

The convolution operation is one of the fundamental building blocks of a convolutional neural network. The convolutional layer's parameters consist of a set of learnable filters (kernels). Every filter is small spatially (along width and height), but extends through the full depth of the input volume. Typical filter sizes might have size 3x3, 5x5, 7x7. The third dimension of the filter corresponds to the number of channels in the input. The grayscale image depth is 1 and the color image has 3 (RGB) color channels.

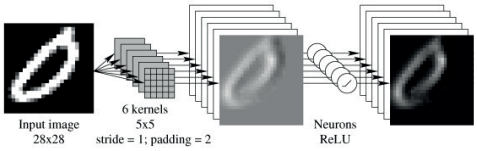

Fig. 1. One convolution layer [18]

During the forward propagation, each filter performs convolution on the input volume across the width and height and compute the dot products between the entries of the filter and the input at any position, this operation is followed by a nonlinear activation function (sigmoid, tanh, ReLU etc.), the resulting outputs are called feature maps. The feature map (also known as an activation map), gives the responses of the filter at every spatial position. An example of convolution layer followed by nonlinear activation is shown in Fig. 1. We stack these activation maps along the depth dimension and produce the output volume. The output volume depends on three hyperparameters: depth, stride and padding.

- The depth of the output volume represents the number of filters that are used in the convolution operation. Each filter is learning something different in the input, edges, blobs, colors.
- The stride is the number of steps that we slide the filter in the input. When the stride is 1 then we move the filters one pixel at a time. When the stride is 2 then the filters jump 2 pixels at a time as we slide them around. This will produce smaller output volumes spatially.
- Padding allows controlling the output size. Applying convolution to an input, reduce the output size that leads to losing information. To avoid that, we pad the input volume with zeros around the border. Two common choices are valid convolution and the same convolution. The valid convolution means no padding, the same convolution means that the output size remains the same as the input size.

The output size is calculated in the following way:

$$(n + 2p - f) / s + 1$$

Where n is the number of filters, p is the amount of padding, f is the filter size and s is the stride.

Pooling layer

CNNs often use pooling layer operation after convolution layers, its function is to reduce the dimension, also referred as subsampling or downsampling. Hyperparameters of pooling layer represent the filter size and strides. Most commonly used pooling layer is with filter size 2 and with stride 2. Two common types of pooling layers are max pooling and average pooling, where the maximum and average value is taken, respectively. Max pooling is used often than average pooling. Pooling layer does not have parameters to learn. The intuition of what max pooling is doing is that the large number means that there may be detected a feature. An example of convolution layer followed by pooling layer is shown in Fig. 2.

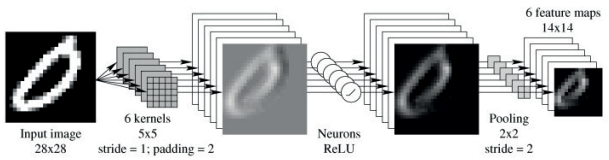

Fig. 2. Convolution layer followed by pooling layer [18]

Fully Connected layer

After several convolution and pooling layers, the CNN generally ends with several fully connected layers. The tensor that we have at the output of these layers is transformed into a vector and then we add several neural network layers. The fully connected layers typically are the last few layers of the architecture as shown in the Fig. 3, the dropout [16] regularization technique can be applied in the fully connected layers to prevent overfitting. The final fully connected layer in the architecture contains the same amount of output neurons as the number of classes to be recognized.

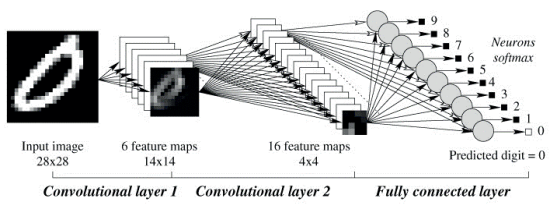

Fig. 3. Two convolutional layers followed by a fully connected layer [18]

3. CLASSIC CONVOLUTIONAL NEURAL NETWORK ARCHITECTURES

We have so far described different layers of CNN. Now, we present how these layers are combined to form the architecture of the network. The most common CNN architectures stack a few convolutional layers together, it follows the pooling layer, then this pattern repeats, and we add at the end the fully connected layers. Classic CNN architectures are LeNet-5 [1], AlexNet [14] and VGGNet [19].

LeNet-5

The LeNet-5 [1] was the first CNN, it proposed by Yann LeCun and his team at Bell Labs in 1998, this architecture is shown in Fig. 4. This network was devoted to digit recognition, LeCun et.al. used the system for handwritten signature detection in checks, and it was successfully deployed commercially for this purpose. It is composed only on few layers and few filters, due to the computer limitations at that time. As shown in Fig. 4, the architecture has two convolution layers, two average pooling layers, two fully connected layers and an output layer with Gaussian connection. LeNet-5 [1] has 60,000 parameters. As activation function, tanh activation function is used.

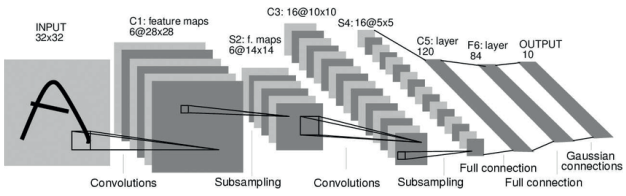

Fig. 4. Architecture of LeNet-5 [1]

AlexNet

The AlexNet [14] architecture was the first work that popularized Convolutional Networks in Computer Vision, it was the winner of the ImageNet ILSVRC [20] competition in 2012, it had a 15.4% top-5 error rate vs 26.2% for the next lowest network. AlexNet [14] follows the pattern of the LeNet-5 [1] architecture, but it was deeper, bigger, and featured convolutional layers stacked on top of each other. The tanh activation function, that was used in LeNet-5 [1], replaced with ReLU

function, as loss function, cross entropy loss function is used. AlexNet [14] used a much bigger training set. LeNet-5 [1] was trained on the MNIST dataset with 50,000 images and 10 categories, AlexNet [14] used a subset of the ImageNet dataset with a training set containing one million color images and 1000 categories.

The input image resolution is 224 × 224, the architecture consists of 5 convolution layers, three 2x2 max-pooling layers and 2 fully connected layers. As shown in Fig. 5, the filter size of the first convolution layer is (11, 11, 3, 96) with a stride of 4, and the first output shape is (55, 55, 96). As a regularization technique, dropout [16] is used in the fully connected layers to reduce overfitting. The total number of parameters in AlexNet [14] is 60 million.

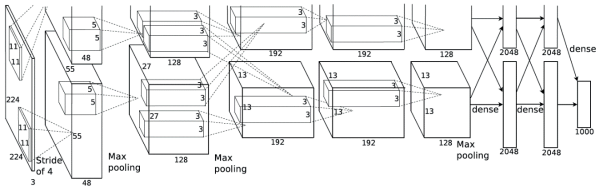

Fig. 5. AlexNet architecture [14]

VGGNet

The VGG Network [19] is introduced in 2014 by Karen Simonyan and Andrew Zisserman. At that time, it was considered as a very deep network. Its main contribution was in showing that the depth of the network is a critical component to achieve better recognition or classification accuracy in CNNs. VGGNet [19] shown in Fig. 6, used 3x3 filters, the authors give the intuition behind this that having two consecutive two 3x3 filters gives an effective receptive field of 5x5, and three 3x3 filters give a receptive field of 7x7 filters. The number of filters in the architecture double after every max-pooling operation.

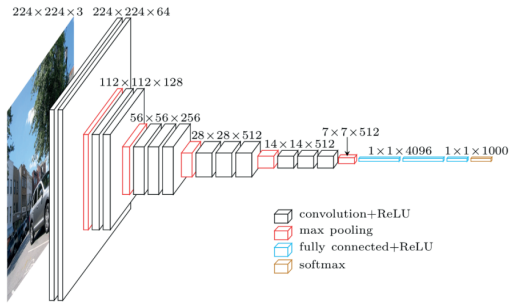

Fig. 6. VGGNet architecture [19]

4. MODERN CONVOLUTIONAL NEURAL NETWORK ARCHITECTURES

Modern CNN architectures are GoogleNet [21], Residual Network [22], Squeeze-and-Excitation network [23]. Each of them is described in more details in the following section.

GoogleNet

The network that won ILSVRC [20] in 2014 is the network GoogleNet [21]. GoogleNet [21], shown in Fig. 8, is considered to be the first use of modern CNN architecture, which is not composed only on successive convolution and pooling layers, it used inception [24] architecture, that is kind of network in network (NIN) [25]. An example is represented in Fig. 7. The inception module skips connections in the network essentially forming a mini-module and that module is repeated throughout the network. The inception module dramatically reduced the number of parameters in the network, GoogleNet [21] employed around 7 million parameters, which represented a 9 times reduction with respect to its predecessor AlexNet [14], which used 60 million parameters. Furthermore, VGGNet [19] employed about 3 times more parameters than AlexNet [14].

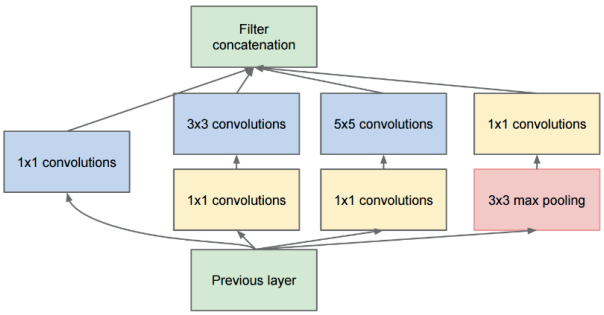

Fig. 7. Inception module [21]

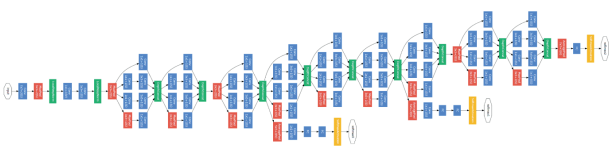

Fig. 8. Inception module [21]
GoogleNet architecture [21]

GoogleNet [21] uses 9 inception modules, and it eliminates all fully connected layers using average pooling to go from 7x7x1024 to 1x1x1024, eliminating a large number of parameters that do not seem to matter much. As a form of data augmentation, multiple crops of the same image were created and the network was trained on it. There are also several followup versions to the GoogleNet [21], most recently Inception-v4 [24].

Residual network

Residual network (ResNet) [22] was the winner of ILSVRC [20] 2015, it has in total of 152 layers. ResNet is built of a residual block, which is shown in Fig. 11, by stacking residual blocks together, each residual block has two 3x3 convolution layer, Periodically, double the number of filters and downsample spatially using stride 2. ResNet [22] features special skip connections and use of batch normalization [26] after every convolution layer. Deeper models are harder to optimize, the solution is to use skip connection, which allows to take the activation from one layer and feed it to another layer. Using that enables to train very deep networks and avoid vanishing and exploding gradient problem. To reduce the number of parameters, the ResNets [22] do not have fully connected layers, besides fully connected layer to output the 1000 classes. ResNet [22] is the first architecture that has better performance than human performance.

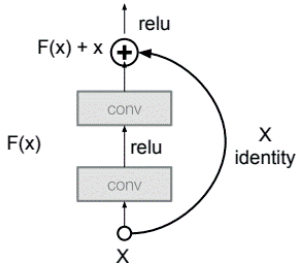

Fig. 9. Residual block [22]

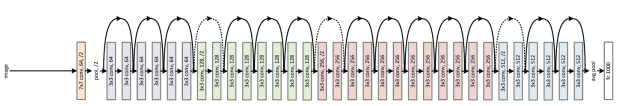

Fig. 10. ResNet [22]

Squeeze-and-Excitation Network

Squeeze-and-Excitation Network (SENet) [23] won the first place on ImageNet challenge in 2017 and significantly reduced the top-5 error to 2.251%. SENet introduce a building block for CNNs that improves channel interdependencies at a minimal additional computational cost. Besides significant improvements in performance, it can be easily added to an existing architecture. The SE block tries to use global information to selectively emphasize informative features and suppress less useful once by adding parameters to each channel of a convolutional block so that the network can adaptively adjust the weighting of each feature map. SE block consists of two operations, squeeze and excitation respectively. Squeeze-and-excitation block is shown in Fig. 13.

- The squeeze operation squeezes each channel to a single numeric value by using global average pooling to generate channel-wise statistic and reduce the dimension.

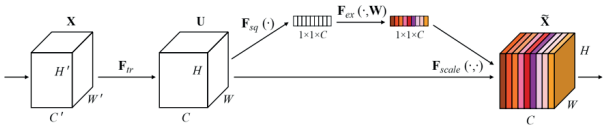

Fig. 11. Squeeze-and-Excitation block [23]

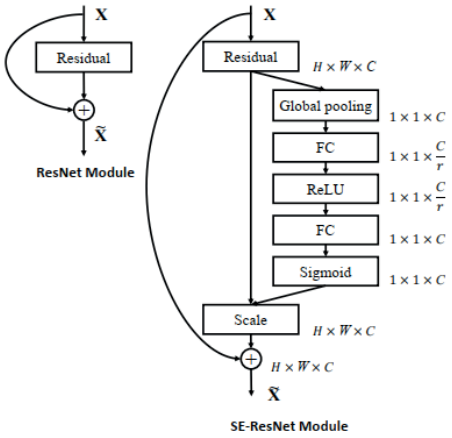

Fig. 12. The schema of the original Residual module (left) and the SEResNet module (right) [23]

- The excitation operation makes use of the information aggregated in the squeeze operation, determines which of the feature maps are important. This is done using a two FC layer around the ReLU non-linearity and in the end the as a gating mechanism, the sigmoid function is applied. The resulting weights applied to each feature maps to generate the output of the SE block which can be fed directly into subsequent layers of the network.

5. CONCLUSION AND DISCUSSION

This paper has outlined the basic concepts of Convolutional Neural Networks, explaining the layers required to build it and detailing how to best structure the network in most image classification tasks. CNN is better than other deep learning methods in applications to computer vision, it gives the best performance in image recognition problems and even outperforms humans in certain cases. The inception module, along with residual networks, has improved CNN performance and introduced new capabilities. The Inception module provides some scale invariance, while residual networks allow training deeper networks. The default choice of network architecture is ResNet or SENet.

REFERENCES

[1] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, 86(11):2278–2324, 1998, pp. 1-46

[2] C. Farabet, C. Couprie, L. Najman and Y. LeCun, "Learning Hierarchical Features for Scene Labeling," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, 2013. pp. 1915-1929

[3] Alex Krizhevsky, Geoffrey E Hinton, “Using very deep autoencoders for content-based image retrieval,” ESANN, 2011, pp. 1-7

[4] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, 2017, pp. 1137-1149

[5] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3156-3164

[6] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1701-1708

[7] A. Toshev, Ch. Szegedy, “DeepPose: Human Pose Estimation via Deep Neural Networks,” Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1653-1660

[8] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale Convolutional Networks," The 2011 International Joint Conference on Neural Networks, San Jose, CA, 2011, pp. 2809-2813

[9] Y. Le Cun and Y. Bengio, “Convolutional networks for images, speech, and time series,” in The Handbook of Brain Theory and Neural Networks, M. A. Arbib, Ed. Cambridge, MA: MIT Press, 1995, pp. 255–258

[10] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." arXiv preprint arXiv:1508.06576, 2015, pp. 1-16

[11] Hubel, D. H. and Wiesel, T. N., “Receptive fields of single neurons in the cat’s striate cortex,” Journal of Physiology, 1959, pp. 574–591

[12] Fukushima, K., “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” Biological Cybernetics, 1980, pp. 193–202

[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard,

W. Hubbard, and L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” in NIPS, 1989, pp. 1-9

[14] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, 2012, pp. 1097–1105

[15] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines” In ICML, 2010, pp. 807-814

[16] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. “Dropout: a simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research, Volume 15, 2014, pp. 1929–1958

[17] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," nature 521.7553, 2015, pp. 436

[18] J. F. Couchot, R. Couturier, C. Guyeux, M. Salomon, “Steganalysis via a Convolutional Neural Network using Large Convolution Filters,” 2016, pp. 1-8

[19] Simonyan, Karen and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv:1409.1556v6, 2015

[20] Russakovsky, Olga et al., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision 115, 2015, pp. 211-252

[21] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going Deeper with Convolutions," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778

[23] Hu, Jie et al, "Squeeze-and-Excitation Networks," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132-7141

[24] Szegedy, S Ioffe, V Vanhoucke, AA Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp 4278-4284

[25] M. Lin, Q. Chen, and S. Yan., "Network in Network" arXiv:1312.4400, 2013

[26] Sergey Ioffe, Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," International Conference on Machine Learning, 2015, pp. 448-456